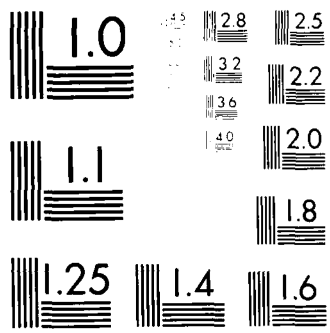
AD-A170 023 AN O(/V/+/E/) ALGORITHM FOR FINDING AN EDGE-MAXIMAL SUBGRAPH WITH A TR-FO . (U) CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENCES RESEAR . E BALAS 1/1

UNCLASSIFIED JUL 85 MSRR-519 N00014-82-K-0329 F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A170 023

AN $O(|V| + |E|)$ ALGORITHM FOR FINDING AN EDGE-MAXIMAL SUBGRAPH WITH A TR-FORMATIVE COLORING

by

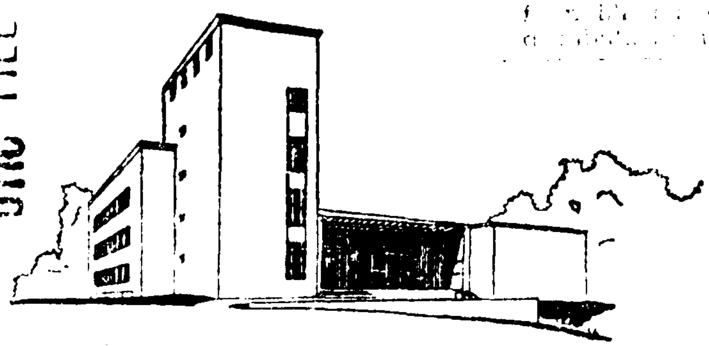
Egon Balas

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



W.P.#8-85-86

Management Science Research Report No. MSRR-519

AN O(|V| + |E|) ALGORITHM FOR FINDING AN EDGE-MAXIMAL SUBGRAPH WITH A TR-FORMATIVE COLORING

by

Egon Balas

July 1985

The research underlying this report was supported by Grant ECS-8503192 of the National Science Foundation and Contract N00014-85-K-0198 with the U.S. Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

Management Science Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

If TR is the class of triangulated graphs, a TR-formative edge coloring is a green/red coloring of the edges of a graph, such that the green graph is triangulated (i.e. belongs to TR) and the red graph has no triangles. Recently Balas, Chvatal and Nesetril gave an $O(|V|^5)$ algorithm for finding a maximum-weight clique in any graph $G = (V,E)$ with a known TR-formative edge coloring. In this paper we give an $O(|V| + |E|)$ time algorithm for finding in an arbitrary graph an edge-maximal subgraph with a TR-formative coloring. This can be used to construct improved implicit enumeration procedures for finding a maximum-weight clique in an arbitrary graph.

Our algorithm consists of two subroutines, also of interest in their own right: one finds an edge-maximal triangulated subgraph, the other one an edge-maximal triangle-free subgraph, in an arbitrary graph.

1. Introduction

Let TR denote the class of triangulated graphs, i.e. graphs that have no hole (chordless cycle of length ≥ 4). For an arbitary graph $G = (V,E)$, we define a TR-formative edge coloring of G as a green/red coloring of the edges of G, such that the green graph is triangulated and red graph has no triangles. In a recent paper, Balas, Chvatal and Nesetril [1985] have given an $O(|V|^5)$ algorithm for solving the maximum weight clique problem (MWCP for short) on any graph G with a known TR-formative edge coloring. This result can be used to derive improved implicit enumeration algorithms for solving the MWCP on an arbitary graph G, provided one has an efficient way of generating subgraphs of G with a TR-formative edge coloring. Indeed, suppose that for some $E' \subset E$, the subgraph $G[E'] = (V, E')$ of G generated by E' has a known TR-formative edge coloring. Then the MWCP on $G[E']$ is solvable in $O(|V|^5)$ time. Let K be a maximum weight clique of $G[E']$ with weight $w(K)$. If G has a clique K' with $w(K') > w(K)$, then $G(K')$ has at least one edge in $E \setminus E'$. Thus an implicit enumeration algorithm can be constructed that, as a branching rule, recursively replaces the current graph G by the collection of subgraphs $G(N(e_1))$, $G(N(e_2)) - e_1, \ldots,$ $G(N(e_q)) - \{e_1,\ldots,e_{q-1}\}$, where $\{e_1,\ldots,e_q\} = E \setminus E'$, and for any edge $e = (u,v)$,

$$N(e) := \{w \in V \setminus \{u,v\} \mid (w,u) \in E \text{ and } (w,v) \in E\},$$

while $G(N(e))$ denotes the subgraph of G induced by $N(e)$. The procedure for finding a subgraph with a TR-formative edge coloring can then be applied to each of the above graphs, while any graph $G(N(e_i)) - \{e_1,\ldots,e_{i-1}\}$ that can be shown to have no clique of weight larger than $w(K) - w(u_i) - w(v_i)$ (where $(u_i, v_i) = e_i$), can be discarded. Naturally, the larger the subgraph with a TR-formative coloring that one is able to generate, the fewer branches are

needed; hence one is moved to search for edge-maximal subgraphs, where we define the latter as follows. A triangulated subgraph G[F] is called <u>edge-maximal</u> (with respect to set inclusion) if there exists no $F' \supsetneq F$, $F' \subseteq E$, such that G[F'] is trangulated. An edge-maximal triangle-free subgraph is defined analogously. Now if G[E'] is a subgraph of G with a TR-formative edge coloring [F,D], where G[F] is triangulated, G[D] is triangle-free, and $F \cup D = E'$, we say that G[E'] is <u>edge-maximal</u> if there exists no $F' \supsetneq F$, $F' \subseteq E \setminus D$, such that G[F'] is triangulated, <u>and</u> no $D' \supsetneq D$, $D' \subseteq E \setminus F$, such that G[D'] is triangle-free.

An implicit enumeration procedure of the above type was used by Balas and Yu [1984] to find a maximum (unweighted) clique in an arbitrary graph. The Balas-Yu algorithm generates a maximal induced subgraph whose chromatic number is equal to the size of its maximum clique, and in which a maximum clique can be found in $O(|V|+|E|)$ time. If G(S) denotes the maximal induced subgraph generated, then G is replaced by $G(N(v_1))$, $G(N(v_2)-\{v_1\}),\ldots,G(N(v_p)-\{v_1,\ldots,v_{p-1}\})$. The computational results obtained on randomly generated graphs with up to 400 vertices and 30,000 edges indicate that the procedure is clearly superior to earlier algorithms that use straight implicit enumeration.

In this paper we give an $O(|V|+|E|)$ procedure for finding an edge-maximal subgraph with a TR-formative coloring in an arbitrary graph. The procedure consists of two independent algorithms, each of which is also of interest in its own right. The first one finds an edge-maximal triangulated subgraph, the second one finds an edge-maximal triangle-free subgraph, in an arbitrary graph. Applied to a graph G = (V,E), algorithm I finds an edge-maximal triangulated subgraph G[F] of G. Applied to $G[E \setminus F]$, algorithm II then finds an edge-maximal triangle-free subgraph G[D] of $G[E \setminus F]$. The resulting graph $G[F \cup D]$ is then an edge-maximal subgraph of G with the TR-formative edge coloring [F,D].

2. Edge-Maximal Triangulated Subgraphs

Our algorithm for finding an edge-maximal (EM) triangulated subgraph of a graph is based on the Balas-Yu algorithm for finding a maximal triangulated induced subgraph. Like the latter, it uses the ideas of a procedure by Rose, Tarjan and Lueker [1976] for testing triangularity, and it runs in $O(|V|+|E|)$ time. An earlier algorithm for finding an EM triangulated subgraph, by Dearing, Shier and Warner [1984[, requires $O(\Delta \cdot |E|)$ steps, where Δ is the maximum degree.

To explain the algorithm, we have to recall a few properties of triangulated graphs. A vertex is called simplicial if all its neighbors are adjacent to each other. Every triangulated graph has a simplicial vertex (Dirac [1961]); it follows that a triangulated graph has at most as many cliques as vertices. An ordering $v_1,\ldots,v_n$ of the $n = |V|$ vertices of a graph $G = (V,E)$ is called perfect if for $i=1,\ldots,n$, v_i is simplicial in $G(\{v_i,v_{i+1},\ldots,v_n\})$. A graph is triangulated if and only if there exists a perfect ordering of its vertices. Based on this property, triangulated graphs can be recognized, and their cliques can be listed, in $O(|V| + |E|)$ time. (Rose, Tarjan and Lueker [1976]).

Given a graph $G = (V,E)$, our algorithm generates an EM triangulated subgraph $G[F]$ of G, along with a perfect ordering σ of the vertices of $G[F]$. The ordering σ is generated backwards (i.e. the last rank is assigned first, the next to last seconds, etc.), based on a lexicographic labeling of the vertices. For some ordering $\sigma = (v_1,\ldots,v_n)$ of the vertices of G, we say that v_j is a successor of v_i if v_j and v_i are adjacent and $j > i$. If, in addition, $k > j$ for all successors v_k of v_i other than v_j, then we say that v_j is the first successor of v_i.

We will use two known facts, established in Rose, Tarjan and Lueker [1976]:

Fact 1. If for $i = 1,\dots,n$, the first successor of v_i is adjacent in $G[F]$ to every other successor of v_i, then for $i = 1,\dots,n$, v_i is simplicial in the subgraph of $G[F]$ induced by $\{v_i,\dots,v_n\}$.

Fact 2. $G[F]$ is an EM triangulated subgraph of G is and only if there exists no $e \in E \setminus F$ such that $G[F \cup \{e\}]$ is triangulated.

Algorithm I (for finding an EM triangulated subgraph)

0. Initialization. Assign the label $\emptyset$ to every vertex. Set $i \leftarrow n = |V|$, $F \leftarrow \emptyset$, $\sigma \leftarrow \emptyset$.

1. Choosing a vertex. If $i = 0$, stop: $G[F]$ is an EM triangulated subgraph of G and σ is a perfect ordering in $G[F]$.

Otherwise, choose an unnumbered (i.e. unranked) vertex v with lexicographically largest label. Assign number i to v, i.e. set $v_i \leftarrow v$, insert v_i into σ as the first element, i.e. set

$$\sigma \leftarrow (v_i, v_{i+1}, \dots, v_n),$$

and go to 2.

2. Adding edges to F. Let $S^{\sigma}(v_i) = (w_1,\dots,w_\ell)$ be the list of successors of v_i. Add to F the edge (v_i,w_1) and all the edges (v_i,w_j) such that w_j is a successor of v_i, adjacent to w_1 in $G[F]$; i.e., set

$$F \leftarrow F \cup (v_i,w_1) \cup \{(v_i,w_j) \mid w_j \in S^{\sigma}(v_i) \text{ and } (w_1,w_j) \in F\}$$

and go to 3.

3. Labeling. Append i to the label of each unnumbered vertex w adjacent to v_i in G, set $i \leftarrow i - 1$, and go to 1.

Theorem 1. Algorithm I generates an EM triangulated subgraph of G in $O(|V| + |E|)$ time.

Proof. Let $\sigma = (v_1,\dots,v_n)$ be the ordering generated by Algorithm I. From Fact 1, for $i = 1,\dots,n$, v_i is simplicial in the subgraph of $G[F]$ induced by $\{v_i, v_{i+1},\dots,v_n\}$. Hence $G[F]$ is triangulated. From Fact 2, to

prove that G[F] is edge-maximal it is sufficient to show that for all $e \in E \setminus F$, $G[F \cup \{e\}]$ is not triangulated. But this follows from the rule which governs the addition of new edges to F (step 2). Thus G[F] is an EM triangulated subgraph of G.

Next we establish the complexity of the algorithm. Each application of Step 3 (labeling) requires $O(\deg v_i)$ operations, and the list of labeled vertices can be kept in lexicographically decreasing order. Thus Step 1 (choosing an unnumbered vertex with lexicographically largest label) can be carried out in constant time. Step 2 (adding edges to F) again requires $O(\deg v_i)$ operations. To get the complexity of the entire algorithm, we sum over all v_i, $i = 1, \ldots, n$, to obtain $O(\sum_{i=1}^{n} \deg (v_i)) = O|V| + |E|)$.

3. Edge-Maximal Triangle-Free Subgraphs

A straightforward method for finding an EM trinagle-free subgraph G[D] of a given graph $G = (V,E)$ is to initialize the edge set D as empty, then examine every edge of E in some arbitrary order and put it into D if and only if this does not create a triangle in D. Examining every edge from this point of view requires $O(|V|)$ steps, hence the whole procedure requires $O(|V| \cdot |E|)$ steps.

However, one can do better than this naive approach by arbitrarily choosing a vertex v_o, partitioning the vertex set V into subsets $V_1,\ldots,V_p$, where V_d is the set of vertices at distance d from v_o (distance being measured by the number of edges in a shortest path), and finally deleting all the edges joining vertices of the same subset V_d, for $d = 1,\ldots,p$. It is not hard to see that the resulting subgraph is edge-maximal triangulated, and its construction takes $O(|V| + |E|)$ steps. The details follow.

For a graph $G = (V,E)$, and a vertex of $v \in V_d$, let

$$N(v) := \{u \in V \mid (u,v) \in E\}; \text{ and}$$

$$N(v)_{d+1} := N(v) \setminus (V_d \cup V_{d-1}) \text{ (with } V_{-1} := \emptyset).$$

W.ℓ.o.g., we assume that G is connected.

Algorithm II (for finding an EM triangle-free subgraph)

0. Initialization. Choose some $v_* \in V$ and set $d \leftarrow 0$, $V_o \leftarrow \{v_*\}$, $V \leftarrow V \setminus \{v_*\}$, $V_1 \leftarrow \emptyset$, $D \leftarrow \emptyset$, and go to 1.

1. Choose the next vertex. If all $v \in V_d$ are marked and $V = \emptyset$, stop: G is an EM triangle-free subgraph of G. If all $v \in V_d$ are marked but $V \neq \emptyset$, set $d \leftarrow d + 1$, $V_{d+1} \leftarrow \emptyset$, and go to 1.

Otherwise choose an unmarked $v \in V_d$, set $v_o \leftarrow v$, and go to 2.

2. Add new vertices and edges. Generate $N(v_o)_{d+1}$, set $V_{d+1} \leftarrow V_{d+1} \cup N(v_o)_{d+1}$, $V \leftarrow V \setminus N(v_o)_{d+1}$, $D \leftarrow D \cup (v_o, N(v_o)_{d+1})$ and go to 1.

Theorem 2. Algorithm II generates an EM triangle-free subgraph in $O(|V| + |E|)$ time.

Proof. For any triangle of any graph and any fixed vertex v_*, two of the three vertices of the triangle are equidistant from v_*; hence the edge joining them is not included into D by the Algorithm. Thus G[D] is triangulated. Further, the only edges of E not in D are those joining pairs of nodes equidistant from v_* and having a common neighbor; but adding any such edge would create a triangle. Thus G[D] is maximal.

To see the complexity of the algorithm notice that for each vertex v_o chosen in step 1, it takes $O(\deg(v_o))$ steps to identify $N(v_o)_{d+1}$ and to add to D the edge set $(v_o, N(v_o)_{d+1})$. Since everything else done in the algorithm takes constant time, the total number of steps is obtained by adding up $O(\deg(v))$ for all $v \in V$, which yields $O(|V|+|E|)$, as claimed.||

4. An Example

Consider the complete 4-partite graph $G = K_{3,3,3,3}$, with 3 vertices in each of its 4 independent sets, numbered $\{1,2,3\}$, $\{4,5,6\}$, $\{7,8,9\}$, $\{10,11,12\}$.

We first apply Algorithm I to G and construct the EM triangulated subgraph G[F] shown in Fig. 1. The labeling is illustrated in Fig.2, where the circled numbers are the ranks (in the ordering σ) and the square brackets contain the lexicographic labels.

Algorithm I.

Iteration 1. Since all labels are $\emptyset$, we choose arbitrarily $v_n = v_{12} \leftarrow 1$. We set $\sigma \leftarrow (1)$ and we append 12 to the label of vertices 4,...,12 (i.e., each of these vertices now have label $L(j)=[12]$, $j=4,\ldots,12$).

Iteration 2. We choose $v_{11} \leftarrow 4$, set $\sigma \leftarrow (4,1)$, $F \leftarrow \{(4,1)\}$; and append 11 to the label of vertices 2,3,7,...,12 (i.e., we set $L(2)=L(3) \leftarrow [11]$, $L(7)= \ldots =L(12) \leftarrow [12,11]$).

Iteration 3. We choose $v_{10} \leftarrow 7$ (as 7 is one of the vertices with lexicographically largest label), set $\sigma \leftarrow (7,4,1)$, $F \leftarrow \{(4,1), (7,1), (7,4)\}$, and append 10 to the label of the vertices 2,3,5,6,10,11,12.

In the next 3 iterations we choose $v_9 \leftarrow 10$, $v_8 \leftarrow 11$ and $v_7 \leftarrow 12$. At the end of iteration 6 we have $\sigma = (12,11,10,7,4,1)$ and $F = \{(4,1), (7,1), (7,4), (10,7), (10,4), (10,1), (11,7), (11,4), (11,1), (12,7), (12,4), (12,1)\}$.

So far every application of step 2 has resulted in the addition to F of all the edges joining v_i to its successors. At the next iteration this situation changes.

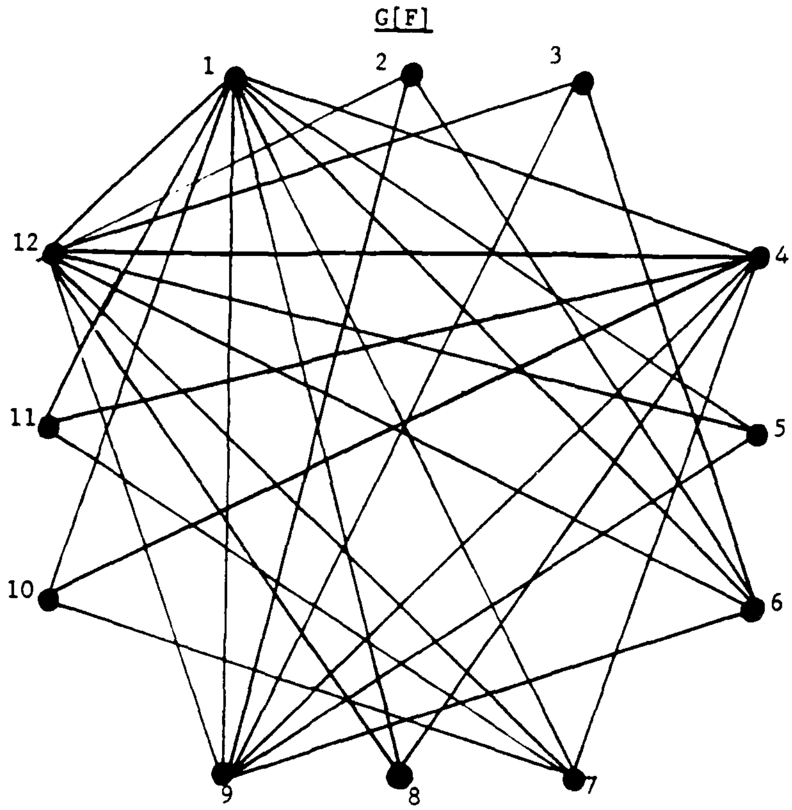

G[F]
1
2
3
4
5
6
7
8
9
10
11
12

Fig. 1

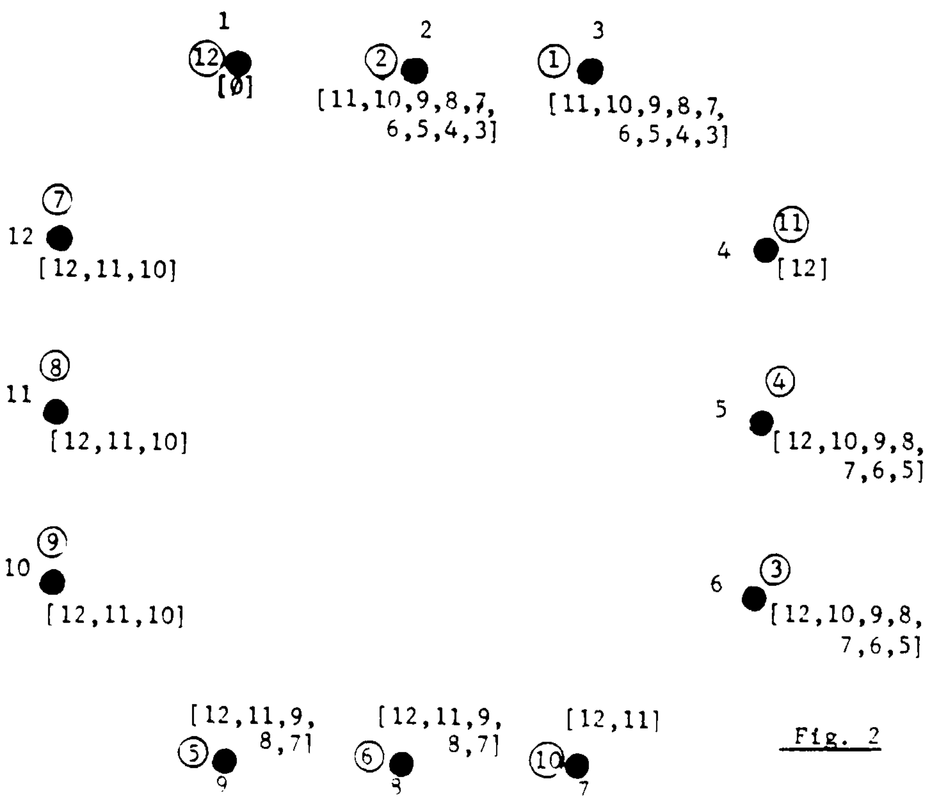

1
12
[Ø]
2
2
[11,10,9,8,7,
6,5,4,3]
3
1
[11,10,9,8,7,
6,5,4,3]
7
12
[12,11,10]
4
11
[12]
8
11
[12,11,10]
5
4
[12,10,9,8,
7,6,5]
9
10
[12,11,10]
6
3
[12,10,9,8,
7,6,5]
[12,11,9,
8,7]
5
9
[12,11,9,
8,7]
6
8
[12,11]
10
7

Fig. 2

$G[E \setminus F]$

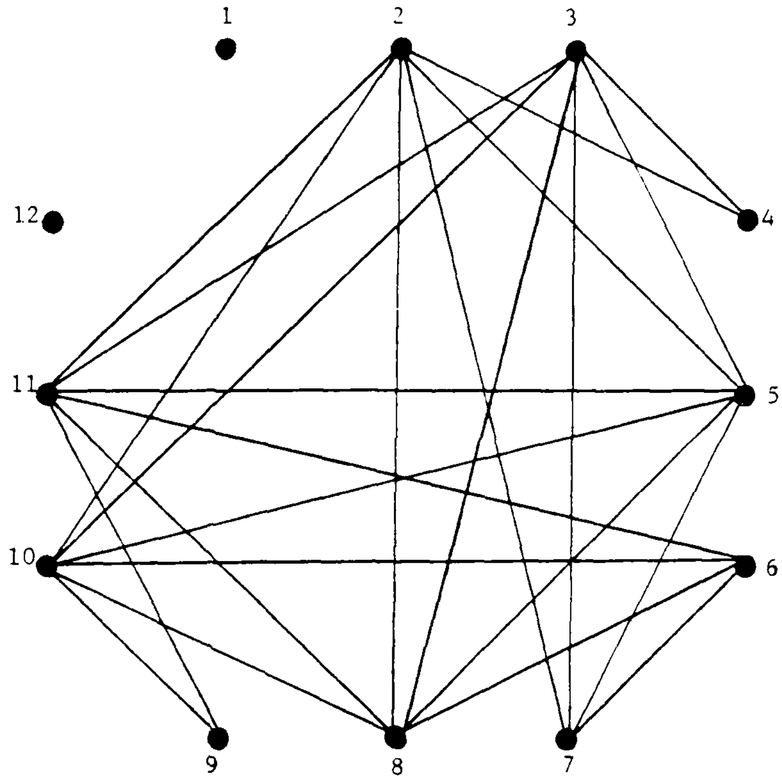


Fig. 3

$G[D]$

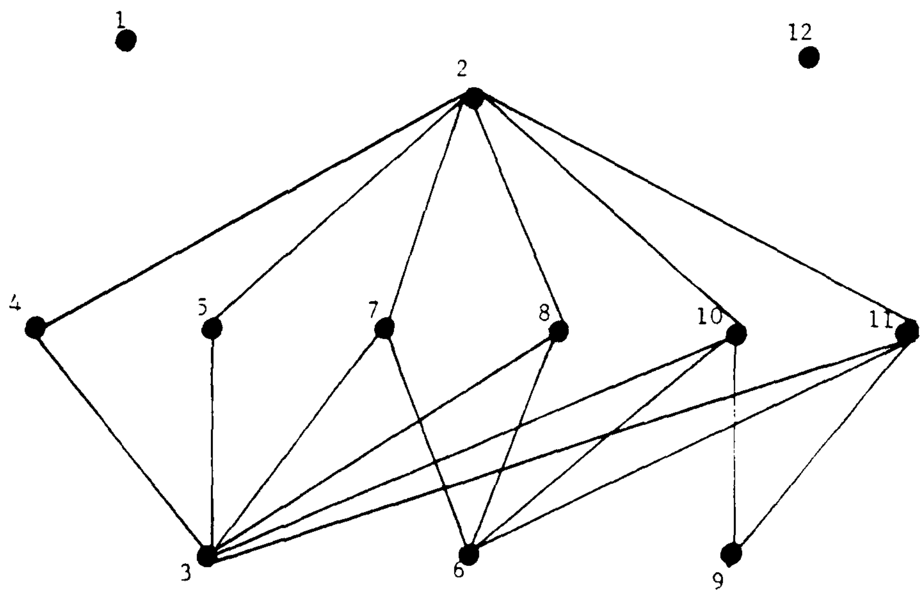


Fig. 4

Iteration 7. We choose $v_6 \leftarrow 8$, since vertex 8 has the lexicographically maximal label [12,11,9,8,7]. We set $\sigma \leftarrow (8,12,11,10,7,4,1)$. The first successor of 8 is 12, so we add (8,12) to F; but of the remaining successors of 8 only 4 and 1 are adjacent in G[F] to 12, hence only (8,4) and (8,1) are added to F, while the remaining two edges, (8,11) and (8,10), remain in $E \setminus F$. Thus we set $F \leftarrow F \cup \{(8,12), (8,4), (8,1)\}$ and append 6 to the labels of all unnumbered vertices adjacent to 8 in G.

Iteration 8. $v_5 \leftarrow 9$, $\sigma \leftarrow (9,8,12,11,10,7,4,1)$, $F \leftarrow F \cup \{(9,12), (9,4), (9,1)\}$. The edges (9,11), (9,10) remain in $E \setminus F$.

Iterations 9,10,11,12 produce $\sigma = (3,2,6,5,9,8,12,11,10,7,4,1)$ and the set of edges shown in Fig. 1.

Next we apply Algorithm II to the graph $G[E \setminus F]$, shown in Fig. 3, to find an EM triangle-free subgraph. We omit the isolated vertices 1 and 12, which can be added to the triangle-free graph obtained. (In general, if $G[E \setminus F]$ is disconnected, we apply the algorithm to every component that contains a triangle).

Algorithm II.

We arbitrarily choose $v_* = 2$ and set $V_o \leftarrow \{2\}$, $V \leftarrow \{3,\dots,11\}$.

Iteration 1. We choose $v_o = 2$ and mark (underline) 2: $V_o \leftarrow \{\underline{2}\}$. We generate $N(2)_1 = \{4,5,7,8,10,11\}$ and set $V_1 \leftarrow N(2)_1$, $V \leftarrow V \setminus V_1 = \{3,6,9\}$, and $D \leftarrow \{(2,4), (2,5), (2,7), (2,8), (2,10), (2,11)\}$.

Iteration 2. $v_o \leftarrow 4$, $V_1 \leftarrow \{\underline{4},5,7,8,10,11\}$, $N(4)_2 \leftarrow \{3\}$, $V_2 \leftarrow \{3\}$, $V \leftarrow \{6,9\}$, $D \leftarrow D \cup \{(3,4)\}$.

Iteration 3. $v_o \leftarrow 5$, $V_1 \leftarrow \{\underline{4},\underline{5},7,8,10,11\}$, $N(5)_2 \leftarrow \{3\}$, $D \leftarrow D \cup \{(3,5)\}$.

Iterations 4 and 5 generate the sets $N(7)_2 = N(8)_2 = (3,6)$, and add to D the edges (3,7), (6,7), (3,8), (6,8). Iterations 6 and 7 generate $N(10)_2 = N(11)_2 = \{3,6,9\}$, and add to D the edges (3,10), (6,10), (9,10), (3,11), (6,11), (9,11).

Iteration 8. At this point $V_1 = \{\underline{4},\underline{5},\underline{7},\underline{8},\underline{10},\underline{11}\}$, i.e. all $v \in V_1$ are marked, and $V = \emptyset$; hence we stop, with the graph G[D] shown in Fig. 4.

The original graph $G = K_{3,3,3,3}$ had 54 edges. G[F] has 30 edges and G[D] has 15. Thus the edge-maximal subgraph $G[F \cup D]$ with the TR-formative edge coloring [F,D] has 45 edges.||

References

E. Balas, V. Chvátal, and J. Nesetril, "On the Maximum-Weight Clique Problem." MSRR no.518, Carnegie-Mellon University, June 1985.

E. Balas and C.S. Yu, "Finding a Maximum Clique in an Arbitrary Graph." MSRR no. 515, Carnegie-Mellon University, December 1984.

P.M. Dearing, D.R. Shier, and D.D. Warner, "Maximal Chordal Subgraphs." Technical Report #406, Department of Mathematical Sciences, Clemson University, September 1983.

D.J. Rose, R.E. Tarjan, and G.S. Lueker, "Algorithmic Aspects of Vertex Elimination on Graphs." SIAM Journal on Computing, 5, 1976, p. 266-283.

(When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: MSRR-519	2. GOVT ACCESSION NO.: AD A171 023
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED: Technical Report, July 1985
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Egon Balas	8. CONTRACT OR GRANT NUMBER(s): N00014-82-K-0329, NR047-607
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA 15213	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Personnel and Training Research Programs, Office of Naval Research (Code 434), Arlington, VA 22217	12. REPORT DATE: July 1985
	13. NUMBER OF PAGES: 12
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Maximum-Weight Clique
Vertex Packing
TR-Formative Edge Coloring
Edge-Maximal Triangulated Subgraph
Edge-Maximal Triangle-Fee Subgraph

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

If TR is the class of triangulated graphs, a TR-formative edge coloring is a green/red coloring of the edges of a graph, such that the green graph is triangulated (i.e. belongs to TR) and the red graph has no triangles. Recently Balas, Chvatal and Nesetril gave an $O(|V|^5)$ algorithm for finding a maximum-weight clique in any graph $G = (V,E)$ with a known TR-formative edge coloring. In this paper we give an $O(|V| + |E|)$ time algorithm for finding in an arbitrary graph an edge-maximal subgraph with a TR-formative

(cont.)

coloring. This can be used to construct improved implicit enumeration procedures for finding a maximum-weight clique in an arbitrary graph.

Our algorithm consists of two subroutines, also of interest in their own right: one finds an edge-maximal triangulated subgraph, the other one an edge-maximal triangle-free subgraph, in an arbitrary graph.

END

DTIC

8-86